2.5D Dust & Magnet Visualization for Large Multivariate Data

Jan Ole Vollmer
Jürgen Döllner
jan.vollmer@hpi.uni-potsdam.de
Hasso Plattner Institute, Faculty of Digital Engineering, University of Potsdam
Potsdam, Germany

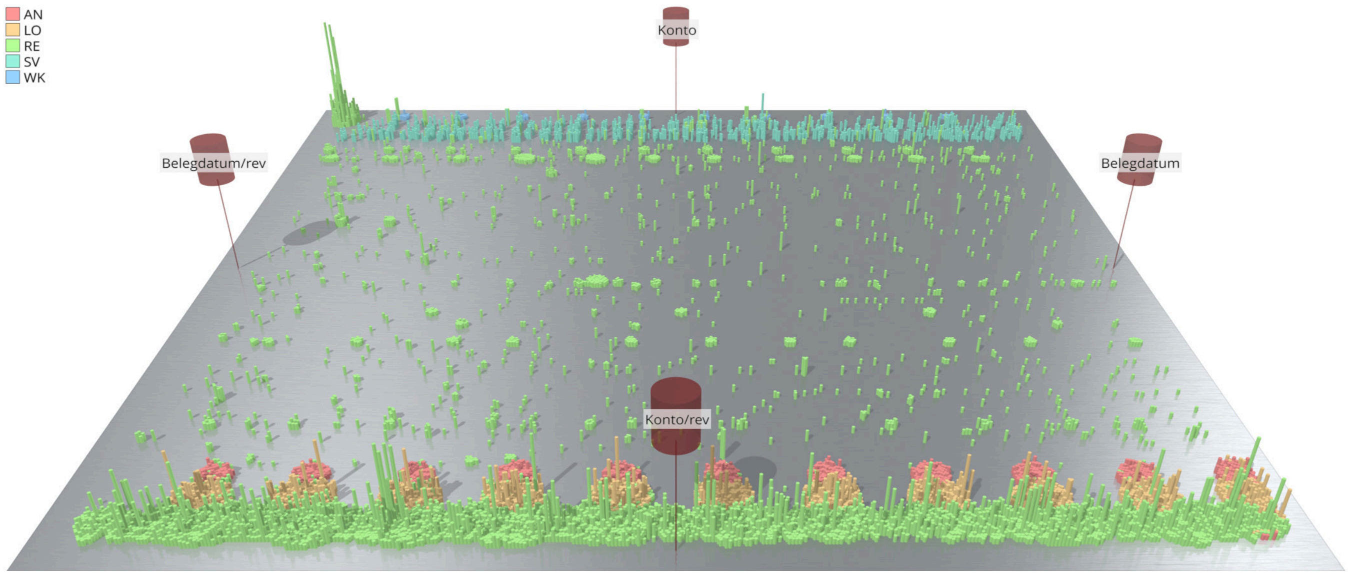

Figure 1: 2.5D dust & magnet visualization of 12,240 particles representing financial transactions of a small company over a year. Spreading out particles by date ("Buchungsdatum", left to right) and account ("Konto", back to front) using a magnet and its inverse (resp.) reveals patterns of recurring bulk transactions clustered at specific accounts and points in time, as well as isolated transactions across the entire range. By additionally mapping the transaction amount to height and source tag to color, the viewer can identify further correlations and exceptions thereof, e.g., red particles in a green cluster (bottom right).

ABSTRACT

In this paper, we present a 2.5D visualization technique based on the dust & magnet metaphor, which generally allows for interactively exploring and analyzing large multivariate data sets. In addition to position and color, we introduce height as additional visual variable for particles to encode extra data attributes in the 2.5D visualization, thus increasing the potential for identifying correlations between attributes. Further, we have developed a real-time collision detection algorithm as part of the particle simulation that ensures overlap-free particle positioning, thereby enabling the continuous perception of patterns, clusters, and outliers. These extensions facilitate on-the-fly validation of hypotheses through the highly dynamic configuration of magnets and visual attribute encoding, which also allows for a better integration of the user's domain knowledge.

We demonstrate the application of our visualization technique using various real-world data sets from different domains, e.g., finance and software analytics.

CCS CONCEPTS

• **Human-centered computing → Visualization techniques**; *Information visualization*; Visualization systems and tools.

KEYWORDS

multivariate visualization, 2.5D, dimensionality reduction, interactive

ACM Reference Format:
Jan Ole Vollmer and Jürgen Döllner. 2020. 2.5D Dust & Magnet Visualization for Large Multivariate Data. In *The 13th International Symposium on Visual Information Communication and Interaction (VINCI 2020), December 8–10, 2020, Eindhoven, Netherlands.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3430036.3430045

VINCI 2020, December 8–10, 2020, Eindhoven, Netherlands

ACM ISBN 978-1-4503-8750-7/20/12.
https://doi.org/10.1145/3430036.3430045

1 INTRODUCTION

The amount of data generated and collected in almost all application fields is rapidly growing as part of digital transformation processes and ubiquitous computational resources, such as in business and finance [27], medicine [16], or the automotive industry [21]. This increase manifests itself in the number of data sets, their size (number of data points), and their dimensionality. In particular, high data dimensionality leads to phenomena such as the *curse of dimensionality* and generally complicates analysis and visualization, for example, in terms of identifying patterns, clusters, and outliers [26].

A wide range of multivariate visualization techniques have been proposed [5, 13] to cope with the limitations of the human visual system. Two general concepts for visualizing high-dimensional data sets are employing *dimensionality reduction* techniques to project high-dimensional data points to a low-dimensional space suitable for common visualization techniques (e.g., scatter plots) while retaining as many similarities and differences as possible, and *visual encoding* where visual attributes such as color, size, or shape are used to encode additional data dimensions [19].

Our work is inspired by the dust & magnet (D&M) metaphor, a 2D visualization technique described by Yi et al. [29] that refers to (iron) dust particles attracted by magnets to position data points on a reference plane. In this metaphor, particles represent data points and magnets represent data dimensions. The attraction of a dust particle is computed for each magnet individually based on the data point's value in the dimension associated with the magnet. Data points with similar attributes are placed together, forming clusters with outliers clearly visible as isolated particles. The viewer can dynamically adjust magnet selection and positions, thus interactively exploring the data set. Visually encoding a further data dimension as particle color enables the user to perceive correlations between this and the magnet dimensions.

Contribution. While the work of Yi et al. focuses on comparing individual particles of small data sets (10s to 100s of elements), we apply the D&M metaphor for the visualization of larger data sets (10,000s to 100,000s of elements), where our primary goal is to support the identification of patterns, clusters, outliers, and correlations, as demonstrated in Figure 1. To this extent, we made the following major contributions:

1. Design and implement an algorithm for overlap-free particle positioning using real-time collision detection that is required to reliably identify large clusters of particles.
2. Adapt and optimize the D&M simulation, rendering, and interaction for large particle quantities of 10,000s to 100,000s.
3. Develop a 2.5D visualization based on the D&M metaphor that uses the particle height to visually encode an additional data dimension, thus enabling the viewer to perceive further correlations. This implies a 3D representation of particles.

The remainder of this paper is structured as follows: In Section 2, we present related work, followed by a description of our physics-based D&M simulation process in Section 3 and the corresponding visualization in Section 4. Finally, in Section 5, we demonstrate the application of our technique using different real-world data sets and discuss its advantages and limitations.

2 RELATED WORK

Numerous techniques and metaphors for visualizing high-dimensional data have been presented (e.g., parallel coordinates, treemaps), as well as a variety of surveys covering various aspects of multivariate visualization [7, 15, 20]. Liu et al. [20] define the visual scalability in terms of either data points or dimensions as one of the major challenges in information visualization and list the question of how to involve the users and their domain knowledge in the data reduction process as interesting research opportunity in this regard. Our proposed technique follows this idea, as the selection and placement of magnets (and by that, data dimensions) is controlled by the user, incorporating their domain knowledge.

2.5D Visualization. The majority of visualization techniques and tools are based on 2D information display, 3D approaches are common for 3D data such as 3D geometry models [4] but also used more and more for visualizing high-dimensional data, for example, in case of software maps [28] or spatiotemporal data [17]. The visualization of multivariate data, in particular, can benefit from 3D portrayal through the additional dimension available for projecting data dimensions. However, occlusion represents a major issue in 3D visualization [8] as it is directly dependent on the view perspective. Perceiving all items in a 3D visualization, therefore, can be difficult (or even impossible) and may require extensive navigation and interaction to gain the desired overview.

The issue can be alleviated by restricting the visualization to "2.5D" [25], denoting portrayal that places data points on a 2D reference plane similar to heightfields. Such an approach retains the ability to map attributes to the third dimension (e.g., by visual encoding as height), while significantly reducing the occlusion problem as items are not stacked in the third dimension. 2.5D visualization is well established for treemaps [3, 18] as well as other areas of scientific and information visualization [2, 10], though not all authors use the term "2.5D". For our visualization, we combine the 2.5D presentation of data points with their arrangement based on the D&M metaphor.

Dust & Magnet. Olson, Korfhage, et al. [22] first described the basic idea of placing *points-of-interest* (POIs) representing data dimensions on a reference plane and then arranging a set of data elements according to individual influences of POIs on the elements. Initially, they focus on arranging a document collection based on the frequency of certain keyword combinations; however, in a subsequent paper, Olsen, Williams, et al. [23] explore other applications as well. Later, Yi et al. [29] generalized the concept for any multidimensional data and introduced the physical metaphor of *dust & magnet.* They target a general audience with little to no experience with multivariate information visualization, thereby emphasizing the intuitiveness and low learning efforts of the physical metaphor for the analysis of small data sets comprising the comparison of individual data points. Our work builds on the D&M metaphor presented by Yi et al.; however, our objective (perceiving clusters and outliers in large data sets) is different from theirs, hence we adapted the simulation process accordingly. Finally, Dai et al. [6] present ongoing research on the potentials of the D&M visualization on large-scale multi-touch displays. We consider their research direction orthogonal to ours with the possibility of combining both.

3 DUST & MAGNET SIMULATION

The D&M metaphor represents data points as dust particles arranged on a 2D reference plane. The user creates a number of magnets, each corresponding to a specific attribute of the data set and places these on the plane, influencing particle positions according to their values for the magnets' attribute. Further, each magnet has a configurable *magnitude* $g \in [-1, 1]$ to scale its influences; setting $g < 0$ results in the magnet repelling dust particles, while $g = 0$ effectively turns off the magnet. In response to changes to a magnet's configuration, i.e., position and magnitude, the system updates dust particle positions in real-time using the simulation algorithm described in Section 3.1. The real-time computation of particle positions is crucial to maintain the impression of directly manipulating particle positions through the interactive arrangement of magnets, which facilitates on-the-fly validation of hypotheses among the multivariate data set. Using this metaphor the user can gain insight through two observations:

- The position of dust particles relative to magnets reveals value distributions (Figure 2) for multiple attributes and combinations thereof.
- The position of dust particles relative to each other shows similarities and differences between particles across multiple attributes, e.g., clusters (group of particles with (nearly) identical attribute values), outliers (single or few isolated particle(s) with one or more attribute values distinct from the majority of particles), and patterns (recurring sets of clusters) as demonstrated in Figure 1.

In general, D&M can be applied to any tabular data set, comprising data points (or elements; rows) with attributes (or dimensions; columns). To compute attractions between magnets and particles, input data must be normalized to $[0, 1]$ prior to simulation, corresponding to zero and maximum attraction, respectively. This is required to prevent particles moving out of the visible area as well as to level attractions across attributes. Normalization is performed for each attribute A (the ordered set of all elements' values for this attribute) separately to account for the different value domains, resulting in $\hat{A}$. This is straightforward for numerical attributes. Likewise, for ordinal attributes, linearly mapping sorted levels to equidistant $x \in [0, 1]$ results in a meaningful interpretation of the attraction; however, applying the same mapping to arbitrarily sorted nominal attributes might result in a false impression of ordering that the user must be aware of. Yet, the assignment of nominal attributes to magnets is an important tool to spread out particles according to categories, as demonstrated in Figure 1.

3.1 Particle Simulation Algorithm

The simulation process determines updated particle positions based on current particle and magnet positions. For the purpose of the simulation, the *attraction* $\alpha \in [-1, 1]$ between a dust particle $p \in P$ and a magnet $m \in M$ is a dimensionless quantity calculated as $\alpha_{p,m} := \hat{x}_{i_p} \cdot g_m$, where $\hat{x}_{i_p} \in \hat{A}_m$ denotes the particle's normalized value for a data attribute associated with the magnet. By inverting the normalized attribute range prior to attraction calculation $\alpha^{-1}_{p,m} := \left(1 - \hat{x}_{i_p}\right) \cdot g_m$, the magnet attraction can be adjusted accordingly, denoted as *inverse magnet*. Note that this differs from a negative magnitude $g = -1$ in that the latter negates attractions, while inverting a magnet does not change the sign of the attraction.

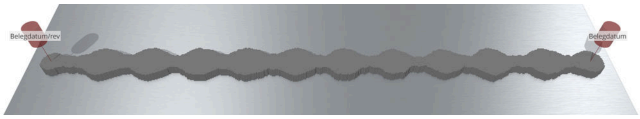

Figure 2: Example of a magnet and its inverse showing the value distribution of a single attribute alternating between intervals with lower and higher density that our collision detection prevents from collapsing.

Yi et al. [29] use a *force-based* algorithm in their work that computes forces exerted by magnets on the particles and then integrates resulting velocities over time They argue that this approach closely matches realistic physical behavior and, therefore, is most intuitive; however, this algorithm suffers from two major drawbacks:

1. Particle positions converge to the magnet with the strongest attraction, making the final result useless. Insights are gained on intermediate simulation steps either through the different particle speeds or through continuously dragging a magnet to perceive its influence on individual particles.
2. As these intermediate steps depend on the entire history of changes to the magnet configuration (and their timing), reproducing the same result to repeat analysis steps is difficult.

We use a *position-based* algorithm that calculates the updated position $\vec{r_p}'$ for a particle p as weighted average of the magnet positions $\vec{r_m}$ according to the individual attractions $\alpha_{p,m}$:

$$\vec{r_p}' := \begin{cases} (0,0), & \text{if } \sum_{m \in M} \alpha_{p,m} = 0 \\ \dfrac{\sum_{m \in M} \alpha_{p,m} \cdot \vec{r_m}}{\sum_{m \in m} \alpha_{p,m}}, & \text{else} \end{cases} \tag{1}$$

As particle positions do not depend on the time, it is sufficient to evaluate this algorithm on-demand upon changes to the magnet configuration. Particle positions are then interpolated between the current and newly-computed final positions across a small number of frames (as described below), since animated transitions are an important tool to convey change information [1]. More specifically, in case of D&M, this allows visual tracking of particles between positioning states.

By directly calculating the final particle positions, this approach avoids the disadvantages of the force-based algorithm at the expense of physically incorrect behavior; however, contrary to Yi et al. [29], we find this approach more intuitive, as magnet influences on particle positions are more directly visible compared to the force-based algorithm. Since the resulting position of a dust particle does not depend on its previous position, the same magnet configuration and weights will always yield the same $\vec{r_p}$, thus solving the reproducibility issue for the core simulation, although the algorithm for overlap-free placement described below can introduce another source of instability.

3.2 Overlap-Free Particle Placement

The particle positions calculated by the simulation algorithm are subject to occlusion, as particles with identical attribute values

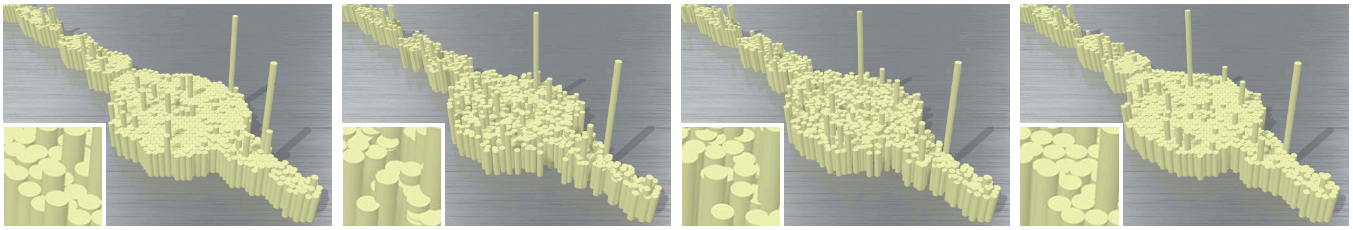

Figure 3: While the initial and final states of a transition between particle positionings are overlap-free, intermediate snapshots are subject to overlaps; however, identification of clusters and perception of their size remains possible.

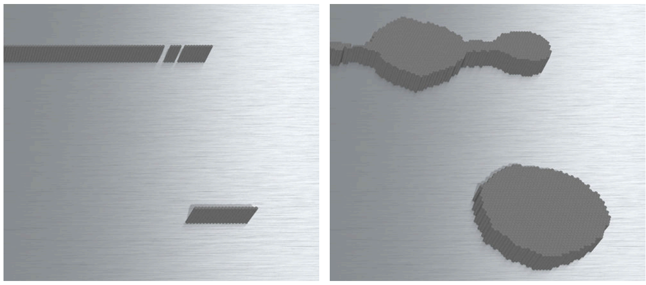

Figure 4: Collision detection between particles retains perceptibility of clusters of particles with identical attribute sets (right) that would otherwise collapse (left).

overlap. While few occurrences of occlusion can be tolerated for small data sets where the goal is to compare individual particles, it prevents the identification of clusters and perception of their sizes for large data sets, as demonstrated in Figure 4. In addition, heights and colors of overlapping particles are difficult to differentiate.

To resolve overlaps, Yi et al. [29] propose a *shake dust*-functionality that pushes particles apart when the user presses a button in the interface. We deem this idea insufficient for our visualization as the perception of clusters is a crucial goal for large data sets and therefore should remain possible throughout the entire process.

For our initial attempt at preventing particles from overlapping, we implemented continuous real-time collision detection for particles using standard rigid-body-physics simulation as well as specialized particle collision detection algorithms [12]. While the former severely limits the number of particles that can be simulated in real-time due the complexity of the simulation intended for complex bodies, the latter can handle the particle counts required for larger data sets in real-time. Using these, particles can move freely across the plane, while showing (to a certain extent) realistic collision behavior (e.g., bouncing); however, both approaches exhibit a major problem: isolated particles are often captured within a group of particles moving in a different direction. As the collision detection prevents the isolated particle from escaping through the surrounding group, it is entrained to a location not corresponding to its attribute values. Consequently, the user might interpret the particle placement incorrectly.

To overcome the captured-particle problem, we slightly weakened the continuous collision detection requirement and designed the following algorithm, the key data structure to which is a hexagonal grid covering the entire reference plane with a cell size matching the particle diameter. The algorithm uses the dust particle position $\vec{r_p}$ computed by the simulation algorithm as *target position* and tries to find a free cell closest to this target position to determine the final placement. Subsequently, the algorithm animates the transition from the current to the final placement over a small number of frames to ensure smooth, continuous movements, during which brief overlaps with other particles may occur. Despite these overlaps, clusters, including their sizes, as well as outliers remain observable during the transition, as demonstrated in Figure 3, which helps the user to track clusters during the interaction.

The algorithm consists of four steps described below. While the optimization and integration steps are executed for every frame, the clearing and insertion steps are only required when the magnet configuration changes.

Insertion. The insertion step is performed whenever target particle positions change. Initially, the grid is cleared by marking all cells as free. Subsequently, each particle p is inserted into the grid at the coordinates $\vec{c_p}$ determined using the following algorithm:

(1) Convert target particle position $\vec{r_p}$ to grid coordinates $\vec{c}$.
(2) Check if cell at $\vec{c}$ is free. If not:
  (a) Generate pseudo-random direction $\vec{d}$ as described below.
  (b) Walk from $\vec{c}$ along $\vec{d}$ until a free cell is found.
  (c) Rotate $\vec{d}$ by 120° and 240° and repeat step 2b for each.
  (d) Of the three cells found during 2a-2c, select the one closest to $\vec{c}$ and update $\vec{c}$ accordingly.
(3) Mark cell at $\vec{c}$ as occupied and return $\vec{c}$.

As the insertion process is repeated whenever target positions change and the insertion uses a random direction, the algorithm potentially suffers from stability issues. To alleviate these, we use the dust particle index i_p as seed for a custom 2D linear congruential generator based on the parameters used in C and Delphi:

$$\vec{z} := \left\lfloor \frac{1}{2^{16}} \left[\left(i_p \cdot \begin{pmatrix} 1103515245 \\ 134775813 \end{pmatrix} + \begin{pmatrix} 12345 \\ 1 \end{pmatrix} \right) \bmod 2^{32} \right] \right\rfloor \tag{2}$$

$$\vec{d} := \operatorname{norm}\left(\vec{z} - 2^{15}\right) \tag{3}$$

The resulting dust particle positions are deterministic and thus reproducible, given identical magnet configurations; however, when using multi-threaded implementation, the order of particle insertions and thus the positions within a cluster become non-deterministic nonetheless, though the sizes and positions of clusters are

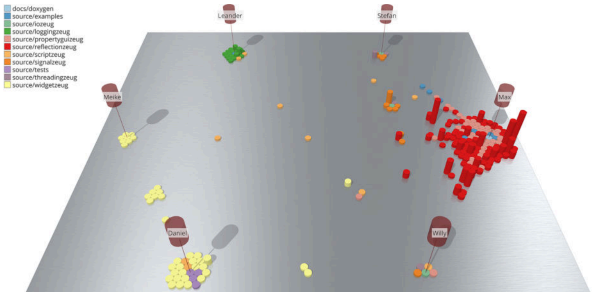

Figure 5: Software implementation responsibilities where particles represent files and magnets represent developers.

stable and reproducible (cf. Section 5.1). Trying three directions and selecting the closest result solves two edge case: (1) Near an edge, finding a free cell along one direction might fail due to all cells being occupied; and (2) when using a single magnet (or a magnet and its inverse), dust particles will form a straight line possibly aligning with the generated direction, in which case the search for a free cell would terminate only beyond the last dust particle and far from the target position.

Searching for free cells along selected lines performs better than search patterns covering an area (e.g., in a spiral) as the number of cells to check grows linearly with the cluster radius, as opposed to quadratically for area-based patterns. Randomly selecting these lines results in an approximately uniform distribution of particles within such a cluster for sufficiently large particle counts, though occasional gaps and unbalanced cluster edges remain.

Optimization. To counterbalance layout defects of the insertion step, particle positions are subsequently optimized using a simple test: for each particle p, take the set of cells adjacent to $\vec{c_p}$ and find one that is free and closer to the particle's target position $\vec{r_p}$ than $\vec{c_p}$. If such a cell $\vec{c_p}'$ exists, set $\vec{c_p} := \vec{c_p}'$. The system performs one update step per frame until no further optimization is possible. As the number of frames required to fully optimize the positioning normally is within the same range as the number of frames required to integrate positions (see below), the optimization step is not noticeable by the user.

Integration. Finally, the integration step performs the animated transition to the newly-assigned position by computing the particle's velocity $\vec{v_p}$ based on the distance between $\vec{c_p}$ and the current position $\vec{q_p}$, as well as a user-configurable speed factor s with: $\vec{v_p} := s(\vec{c_p} - \vec{q_p})$ and then proceeds to integrate the velocity over time: $\vec{q_p} := \vec{q_p} + \Delta t \vec{v_p}$, where Δt denotes the time delta to the previous frame. By re-calculating velocities for each frame, the speed of particle movements gradually decreases, resulting in a smooth transition to the final state, given an appropriate $s \in \left[\frac{0.2}{\Delta t}, \frac{0.5}{\Delta t}\right]$ (approx., determined experimentally). Choosing lower speed factors reduces interactivity, as particle movements in response to changes in the magnet configuration become less direct, while higher speed factors hinder tracking of individual particles during the animation. Setting $s := \frac{1}{\Delta t}$ (assuming constant Δt) effectively disables animation and immediately translates particles to their final positions.

4 DUST & MAGNET VISUALIZATION

In accordance with the theme of magnets and iron dust, dust particle and magnet shapes are placed on a ground plane in a brushed metal look with reflections. The applied surface texturing aids visual distinction of the ground plane, while the reflections provide additional depth cues that improve perception of dust particle positions on the ground plane, as can the additional shadow rendering [14].

4.1 Attribute Encoding

In general, the identification of correlations between attributes provides valuable insights in addition to patterns, clusters, and outliers. While the interactive arrangement of magnets and dust particles is well suited to identify clusters and outliers, the combination of different attributes into a single position prevents the identification of correlations between these attributes. To overcome this limitation, we use height and color as additional visual variables to encode further attributes, enabling the viewer to visually identify correlations between these attributes and the particle locations. Besides visually matching a variable, such as color, to the particle positions with regard to a particular magnet to identify a corresponding correlation, the combination of particle clustering using multiple magnets and visual attribute encoding enables the perception of correlations between clusters and encoded attributes. Figure 5 shows an example of correlating software modules encoded as color and code complexity mapped to height with implementation responsibilities of different developers (cf. Section 5).

Earlier work uses the particle color and radius as visual variables. While we adopt the use of color, we refrained from encoding attributes as particle radius, as this would break the direct relation between a cluster's area and its number of particles and therefore falsify the perception of its size and importance. Instead, we extend the visualization to 2.5D to incorporate the particle height as visual variable, thus retaining the option to simultaneously correlate multiple encoded attributes. The extension to 2.5D requires the 3D rendering of particles and magnets described below, with the inherent problems of 3D portrayals, as discussed in Section 5.1. If a 3D perspective is undesired and height mapping not required, the view could be easily converted to 2D by using an orthographic top-down projection instead of the current perspective projection.

Color Mapping. The dust particle color can encode any attribute type (i.e., numerical, ordinal, nominal), as the system offers appropriate color schemes for all three types. The set of color schemes is customizable to allow integration of schemes according to different sources and requirements. This helps preventing misinterpretations due to a false sense of ordering, e.g., when using a continuous scale for nominal data. The combination of color mapping and lighting may present a problem, as lighting alters the apparent colors and may lead to ambiguity; however, the work of Engel et al. [9] suggests reduced error rates for certain tasks. In our case, lighting may affect value estimation from a shaded fragment and a legend; however, comparison of two or more particles with regard to their colors is not (or less) affected, as due to the homogenous shape and orientation of their representations, the same faces with the same lighting characteristics for each particle are facing the virtual camera in a typical view. In addition, top faces always show the ground truth color.

Height Mapping. As opposed to color, the particle height should not be used to encode nominal attributes, as visually comparing varying heights always yields the impression of one particle being greater than the other and thus more important. We therefore limit the height mapping to numerical and ordinal attributes, where the order relation is well defined. Due to the salience of particle heights and especially outliers demonstrated in Figure 1, they are best used to encode significant attributes of the data set orthogonal to the attributes represented by magnets.

4.2 Visual Representations

Due to the extension of our visualization to 2.5D, our representations of dust particles and magnets must be adapted accordingly.

Dust Particle Representation. The original work of Olsen, Korfhage et al. [22] was designed to visualize document collections, and thus uses a document icon to represent elements, while the general-purpose visualization of Yi et al. [29] represents a particle as circle. For our 2.5D visualization, we extrude this circle to a cylinder. Due to the collision detection, particles are tightly packed within a cluster, raising the question of delineating individual particles. As opposed to simpler shapes, such as a square/cuboid, the most dense packing of circles/cylinders in a hexagonal grid does not require artificial spacing to discern individual particles, thus maximizing particle density at the cost of increased rendering complexity.

The user can adjust the particle radius (linked to the simulation) depending on the current goal and data set, where larger radii are generally used for small data sets to compare individual particles, while smaller radii are suitable for large data sets, reducing the problem of two or more cluster merging due to insufficient space.

Magnet Representation. Magnets are represented by 3D shapes floating above the plane and connected to it using a narrow vertical line. The floating shape avoids the need for collision detection between particles and magnets, while the line marks the exact location on the plane as reference for particle positions. In addition, our grid-based particle placement algorithm limits possible shapes for direct placement on the ground plane to those fitting into a grid cell, which would result in undesired similarities of particle and magnet representations. Semi-transparent rendering of magnets mitigates the issue of magnets occluding particles (cf. Section 5.1).

4.3 Interaction

User interaction is a key element in the data exploration and analysis using our 2.5D D&M visualization. For a comprehensive demonstration see the accompanying video[1].

The interactive placement of magnets is the key process through which knowledge and insights are gained. Our system allows the direct manipulation of magnet positions by dragging their visual representation across the reference plane and updates particle positions accordingly in real-time. By moving one magnet at a time, the user can explore the magnet's attraction for each particle through the speed and distance of their movements in response to the magnet movement. We restrict the possible magnet positions to a limited area corresponding to the rendered ground plane which helps the user in retaining an overview, as it prevents loosing track of isolated magnets and allows the user to return to a well-defined view including the entire visualization anytime. In addition, useful patterns for magnet arrangements identified in Section 5.1 are provided as presets to support a quick start.

While the placement of magnets can only be reasonably performed using an overview perspective, the detailed inspection of specific particle patterns requires a close-up perspective. The user explores the visualization using the standard virtual camera operations pan, rotate, and zoom in a *world-in-hand* navigation metaphor. As the visualization uses a fixed ground plane, camera operations are constrained with regard to this plane, e.g., virtual camera always placed above and upright, which helps preventing navigational errors frequently occurring in 3D [11]. In a typical analysis cycle, the user starts in the default overview perspective by experiment with different magnet arrangements until a suitable configuration is found, and then proceeds to alternate between overview and close-up views to inspect individual clusters.

5 DISCUSSION

We have applied our 2.5D D&M visualization technique for various multivariate data sets from different application domains.

Financial Transactions. The example in Figure 1 visualizes financial transactions, supporting a kind of visual inspection which provides key insights into the financial well-being of a company, while simultaneously constituting a major challenge (due to the large number of elements, as well as attributes) for auditors, e.g., from governmental organizations investigating tax frauds, or private institutions securing their investments. Using our visualization technique with the selected magnet arrangement and attribute encoding, a number of observations can be made:

- Particle clusters corresponding to specific points in time and accounts reveal a recurring pattern of transaction bulks at end of each month.
- Correlating these with the source tag mapped to color shows that some accounts have transactions mostly associated with the same tag (e.g., "WK" in the frontmost row), the exceptions to which may hint transactions of particular interest.
- Mapping of transaction amount to height reveals transactions of exceptional amount with regard to their respective account, hinting transactions of particular interest as well.
- Using the particle color to represent different value-added-tax (VAT) rates, which is a common source for committing tax fraud [24], helps analyzing their validity by correlating VAT rates with accounts, as well as other transactions. This is demonstrated in the accompanying video.

Implementation Responsibilities. While most of the research in the area of software visualization (e.g., using treemaps) targets source code metrics and runtime behavior, our techniques additionally allows analyzing development processes, such as revealing who is responsible for the implementation of modules (Figure 5). We approximate the implementation responsibility by counting the number of modifications (i.e., commits) by developer and source code file and normalizing the result by file, resulting in values expressing, e.g., "x% of modifications to file Z were performed by developer A". By creating a magnet for each developer, dust particles

[1] https://youtu.be/l5DSLij_6rY

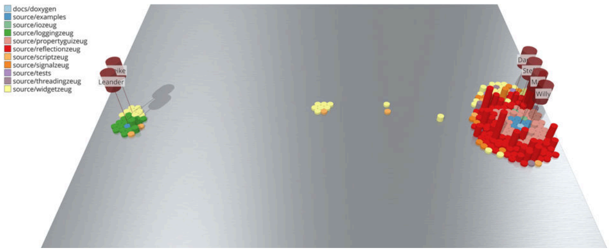

Figure 6: Example of Figure 5 with magnets re-arranged to combine their attractions.

representing source code files are positioned close to their main developer(s), revealing varying implementation responsibilities across developers, e.g., in case of the example, a single developer mainly responsible for the entire project. Mapping of modules to color additionally shows implementation responsibilities by modules, while simultaneously visualizing the module size. Further, mapping of code complexity (e.g., in terms of number of C++-templates) to particle height correlates the complexity with implementation responsibilities and thus developers.

5.1 Advantages and Limitations

Our visualization technique does not include an automated analysis or arrangement of the data common for other multivariate visualization techniques, as one goal of our work is the integration of the user's domain knowledge, which we support through the user-controlled selection and arrangement of magnets. Further, by allowing the direct manipulation of magnets and updating particle positions in response in real-time, we enable the user to interactively develop, and refine hypotheses on-the-fly during the visualization. The drawback of this flexibility is that user interaction is not only possible, but required to create a meaningful visualization, relying on the user to correctly apply the tools provided by the system while respecting their limitations described below.

Magnet Arrangement. The flexible configuration of magnets in terms of attribute mapping and placement opens a wide range of opportunities to explore various characteristics of the data set. During our experiments, certain patterns for arranging magnets proved effective. Most prominently, this is the combination of a magnet and its inverse, i.e., both represent the same attribute, one with unmodified and one with inverse attraction, resulting in particles distributed linearly between both magnets, as shown in Figure 2. By combining two pairs of magnets and their respective inverses and arranging them in the shape of a cross, dust particles are positioned in a rectangular map with the axes corresponding to the two selected attributes, this is demonstrated in Figure 1. This arrangement has similarities to a 2D scatter plot; however, the overlap-free particle placement ensures correct perception of clusters.

The circular arrangement of magnets (Figure 5) is the second common pattern, suitable for an initial overview. Subsequently, the user experiments with replacing and repositioning individual magnets, working towards significant particle arrangements. Placing multiple magnets near each other effectively adds their attractions, while opposite placements subtract attractions, as demonstrated in Figure 6 where the implementation responsibilities shown in Figure 5 are separated by main developers and contributors, according to organizational structure. In this form, the interactive D&M visualization process constitutes a dimensionality reduction technique guided by the user and involving their domain knowledge, the inclusion of which is an important aspect of such techniques [20].

Dust Particle Positioning. The positioning of particles in relation to each other constitutes the main source for insights gained from our visualization; however, there are caveats the user must be aware of. First, the position of a particle is the result of multiple magnets' influences and thus a combination of multiple attributes; its position relative to a particular magnet does not necessarily convey an accurate impression of the particle's absolute value for this attribute as such an isolated inspection disregards influences of other magnets, e.g., in case of the implementation responsibilities example: a particle located near a magnet representing developer *A* does not necessarily mean that the corresponding file was modified particularly often by *A*, but merely that this file was modified more often by *A* than by others. A notable exception to this rule is the cross-shaped arrangement of two pairs of magnets and their inverses; the resulting attractions are orthogonal and therefore separable.

Second, The location of a single particle within a cluster (*micro level*) is not well-defined, more precisely: a particle's position relative to all other particles with identical attributes can be considered non-deterministic. This is not due to the random component of our placement algorithm, but due to the mutual influences of the large number of particles, the result of which is impossible to predict. In case multiple clusters of particles with identical attributes intersect, the positions of particles within the intersection must be considered random as well; however, particle positions observed across multiple such clusters (*macro level*) are meaningful.

Whether these limitations prevent correct interpretation of the visualization depends on the user's task; other visualization techniques are probably more appropriate if exact results are desired. As our goal is the interactive exploration of large data sets primarily though identification of patterns, clusters, and outliers, the limitations discussed above are of minor relevance in this case.

2.5D Visualization. While extending the D&M visualization to 2.5D enhances the perception of correlations among multivariate data items, the required 3D rendering incurs a number of drawbacks. First, user navigation in 3D tends to be more difficult due to the ambiguous meaning of 2D inputs to the 3D camera control—hence we restrict the camera movement as described in Section 4.3.

Second, the problem of occlusion is inherent to any visualization using a 3D portrayal. In our case, two main sources for occlusion must be considered: magnets occluding particles, and particles occluding other particles. We alleviate the issue of magnets occluding particles by rendering magnet shapes semi-transparently, allowing the user to perceive at least the existence of occluded particles, which avoids the most severe consequence of occlusion [8]. The amount of occlusion between particles depends on the viewing angle: for most cases, occlusion is only partial for particles of equal heights, retaining perception of their existence, arrangement, and color; only extreme angles, i.e., near horizontal exhibit full occlusion. When using height mapping, the occlusion problem is more severe as smaller particles may be occluded by taller particles at tilted viewing angles; however, the constant radius limits the number of particles fully occluded by a single particle. In most cases, occluded particles will be partially visible, hence we consider our 2.5D visualization less prone to misinterpretation compared to other 2.5D and 3D visualization techniques, such as treemaps, where a large block can occlude a significant number of smaller elements. Nonetheless, the user can toggle height mapping off altogether.

Technical Limitations. The number of dust particles in a simulation is CPU-bound, as the collision detection proved to be the most resource-intensive task with a mid-ranged desktop CPU being capable of simulating ca. 100,000 particles in real-time. With more powerful CPUs, the number of dust particles may be further increased accordingly.

5.2 Summary and Future Work

In this paper, we presented a visualization technique extending the D&M metaphor to 2.5D; it focuses on supporting interactive analysis and exploration of large multivariate data sets. By ensuring overlap-free particle positioning, our approach enables the continuous perception of patterns, clusters, outliers, and correlations. Future research opportunities include the automatic identification of particle clusters as well as the visualization of time-variant data sets by animating particle positions over time, e.g., the evolution of implementation responsibilities throughout the history of a software project. Further opportunities from the field of software analytics are the refinement of our initial attempt to visualize implementation responsibilities by defining the metric more precisely as well as the exploration of other characteristics not covered by structural and behavioral software analysis.

ACKNOWLEDGMENTS

This work originated from the *GeoPortfolio* research project funded by the German Federal Ministry of Education and Research (BMBF) under the grant number 01IS18010B. The authors are responsible for the content.

REFERENCES

[1] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. 2013. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 20, 5 (May 2013), 740 – 754. https://doi.org/10.1109/TVCG.2013.254

[2] Michael Baur, Ulrik Brandes, Marco Gaertler, and Dorothea Wagner. 2005. Drawing the AS Graph in 2.5 Dimensions. In *Graph Drawing*, János Pach (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 43–48.

[3] Thomas Bladh, David A. Carr, and Jeremiah Scholl. 2004. Extending Tree-Maps to Three Dimensions: A Comparative Study. In *Computer Human Interaction*, Masood Masoodian, Steve Jones, and Bill Rogers (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 50–59.

[4] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and M Eduard Gröller. 2005. Illustrative context-preserving volume rendering. In *EuroVis*. 69–76.

[5] Winnie Wing-Yi Chan. 2006. A survey on multivariate data visualization. *Department of Computer Science and Engineering. Hong Kong University of Science and Technology* 8, 6 (2006), 1–29.

[6] Andrew Dai, Ramik Sadana, Charles D Stolper, and John Stasko. 2015. Hands-On, Large Display Visual Data Exploration. Poster presented at IEEE InfoVis 2015. https://www.cc.gatech.edu/~stasko/papers/infovis15-poster-dnm.pdf

[7] M. C. Ferreira de Oliveira and H. Levkowitz. 2003. From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics* 9, 3 (July 2003), 378–394. https://doi.org/10.1109/TVCG.2003.1207445

[8] Niklas Elmqvist and Philippas Tsigas. 2008. A Taxonomy of 3D Occlusion Management for Visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 5 (Sept. 2008), 1095–1109. https://doi.org/10.1109/TVCG.2008.59

[9] Juri Engel, Amir Semmo, Matthias Trapp, and Jürgen Döllner. 2013. Evaluating the Perceptual Impact of Rendering Techniques on Thematic Color Mappings in 3D Virtual Environments. In *Vision, Modeling & Visualization*, Michael Bronstein, Jean Favre, and Kai Hormann (Eds.). The Eurographics Association. https://doi.org/10.2312/PE.VMV.VMV13.025-032

[10] Paolo Federico, Wolfgang Aigner, Silvia Miksch, Florian Windhager, and Michael Smuc. 2012. Vertigo zoom: combining relational and temporal perspectives on dynamic networks. In *AVI*.

[11] George Fitzmaurice, Justin Matejka, Igor Mordatch, Azam Khan, and Gordon Kurtenbach. 2008. Safe 3D Navigation. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games* (Redwood City, California) *(I3D '08)*. ACM, New York, NY, USA, 7–15. https://doi.org/10.1145/1342250.1342252

[12] Simon Green. 2010. Particle simulation using cuda. *NVIDIA whitepaper* 6 (2010), 121–128.

[13] Patrick E. Hoffman and Georges G. Grinstein. 2002. A Survey of Visualizations for High-dimensional Data Mining. In *Information Visualization in Data Mining and Knowledge Discovery*, Usama Fayyad, Georges G. Grinstein, and Andreas Wierse (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 47–82. http://dl.acm.org/citation.cfm?id=383784.383790

[14] Helen H. Hu, Amy A. Gooch, Sarah H. Creem-Regehr, and William B. Thompson. 2002. Visual Cues for Perceiving Distances from Objects to Surfaces. *Presence: Teleoperators and Virtual Environments* 11, 6 (2002), 652–664. https://doi.org/10.1162/105474602321050758 arXiv:https://doi.org/10.1162/105474602321050758

[15] J. Kehrer and H. Hauser. 2013. Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 19, 3 (March 2013), 495–513. https://doi.org/10.1109/TVCG.2012.110

[16] Joonseok Kim and Peter W. Groeneveld. 2017. Big Data, Health Informatics, and the Future of Cardiovascular Medicine. *Journal of the American College of Cardiology* 69, 7 (2017), 899–902. https://doi.org/10.1016/j.jacc.2017.01.006 arXiv:http://www.onlinejacc.org/content/69/7/899.full.pdf

[17] Menno-Jan Kraak. 2003. The space-time cube revisited from a geovisualization perspective. In *Proc. 21st International Cartographic Conference*. 1988–1996.

[18] Daniel Limberger, Willy Scheibel, Stefan Lemme, and Jürgen Döllner. 2016. Dynamic 2.5D Treemaps Using Declarative 3D on the Web. In *Proceedings of the 21st International Conference on Web3D Technology* (Anaheim, California) *(Web3D '16)*. ACM, New York, NY, USA, 33–36. https://doi.org/10.1145/2945292.2945313

[19] Daniel Limberger, Willy Scheibel, Matthias Trapp, and Jürgen Döllner. 2019. Advanced Visual Metaphors and Techniques for Software Maps. In *Proceedings of the 12th International Symposium on Visual Information Communication and Interaction* (Shanghai, China) *(VINCI '19)*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3231622.3231638

[20] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. 2014. A survey on information visualization: recent advances and challenges. *The Visual Computer* 30, 12 (01 Dec. 2014), 1373–1393. https://doi.org/10.1007/s00371-013-0892-3

[21] A. Luckow, K. Kennedy, F. Manhardt, E. Djerekarov, B. Vorster, and A. Apon. 2015. Automotive big data: Applications, workloads and infrastructures. In *2015 IEEE International Conference on Big Data (Big Data)*. 1201–1210. https://doi.org/10.1109/BigData.2015.7363874

[22] Kai A. Olsen, Robert R. Korfhage, Kenneth M. Sochats, Michael B. Spring, and James G. Williams. 1993. Visualization of a document collection: The vibe system. *Information Processing & Management* 29, 1 (1993), 69–81. https://doi.org/10.1016/0306-4573(93)90024-8

[23] Kai A Olsen, James G Williams, Kenneth M Sochats, and Stephen C Hirtle. 1991. *Ideation Through Visualization: the VIBE System.* Technical Report. University of Pittsburgh, Pittsburgh, PA. http://d-scholarship.pitt.edu/18255/

[24] Grzegorz Poniatowski, Mikhail Bonch-Osmolovskiy, José María Durán-Cabré, Alejandro Esteller-Moré, and Adam Śmietanka. 2018. Study and Reports on the VAT Gap in the EU-28 Member States: 2018 Final Report. http://www.case-research.eu/files/?id_plik=5692

[25] H. Schumann. 2015. 3D in der Informationsvisualisierung. In *Proceedings Go-3D 2015*. https://vcg.informatik.uni-rostock.de/~schumann/papers/2014+/Go-3D.pdf in German.

[26] Tran Van Long. 2010. *Visualizing High-density Clusters in Multidimensional Data*. Ph.D. Dissertation. Jacobs University.

[27] Miklos A. Vasarhelyi, Alexander Kogan, and Brad M. Tuttle. 2015. Big Data in Accounting: An Overview. *Accounting Horizons* 29, 2 (2015), 381–396. https://doi.org/10.2308/acch-51071 arXiv:https://doi.org/10.2308/acch-51071

[28] R. Wettel and M. Lanza. 2007. Visualizing Software Systems as Cities. In *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*. 92–99. https://doi.org/10.1109/VISSOF.2007.4290706

[29] Ji Soo Yi, Rachel Melton, John Stasko, and Julie A. Jacko. 2005. Dust & Magnet: Multivariate Information Visualization Using a Magnet Metaphor. *Information Visualization* 4, 4 (2005), 239–256. https://doi.org/10.1057/palgrave.ivs.9500099